Coherent Configuration の同型計算 (Computer Algebra : Design of Algorithms, Implementations and Applications)

AUTHOR(S):

宮本, 泉

CITATION:

宮本, 泉. Coherent Configuration の同型計算 (Computer Algebra : Design of Algorithms, Implementations and Applications). 数理解析研究所講究録 2007, 1568: 34-39

ISSUE DATE:

2007-09

URL:

http://hdl.handle.net/2433/81231

数理解析研究所講究録
第1568巻 2007年 34-39

Coherent Configurationの同型計算

宮本 泉
IZUMI MIYAMOTO
山梨大学
UNIVERSITY OF YAMANASHI*

1 Coherent Configuration

集合 $\Omega=\{1,2,\cdots,n\}$ とし、$\{R_1,R_2,\cdots,R_d\}$ を $\Omega\times\Omega$ の分割とする。A_k, $k=1,2,\cdots,d$, は $n\times n$ 行列で、

$$(A_k)_{i,j}=\begin{cases}1 & (i,j)\in R_k \text{ のとき}\\ 0 & \text{そうでないとき}\end{cases}$$

とさだめる。これらの行列を下の行列 A でまとめて表す。

$$A=1A_1+2A_2+\cdots+dA_d$$

例 $\Omega=\{1,2,3,4,5,6\}$, $d=7$

$$A=\begin{pmatrix}1&3&4&5&5&5\\4&1&3&5&5&5\\3&4&1&5&5&5\\6&6&6&2&7&7\\6&6&6&7&2&7\\6&6&6&7&7&2\end{pmatrix}$$

$$A_1=\begin{pmatrix}1&0&0&0&0&0\\0&1&0&0&0&0\\0&0&1&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{pmatrix}\quad A_2=\begin{pmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&1&0&0\\0&0&0&0&1&0\\0&0&0&0&0&1\end{pmatrix}\quad A_3=\begin{pmatrix}0&1&0&0&0&0\\0&0&1&0&0&0\\1&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{pmatrix}$$

$$A_4={}^tA_3\quad A_5=\begin{pmatrix}0&0&0&1&1&1\\0&0&0&1&1&1\\0&0&0&1&1&1\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\end{pmatrix}\quad A_6={}^tA_5\quad A_7=\begin{pmatrix}0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&0&0\\0&0&0&0&1&1\\0&0&0&1&0&1\\0&0&0&1&1&0\end{pmatrix}$$

*imiyamoto@yamanashi.ac.jp

定義　$\{R_k\}_{k=1,2,\cdots,d}$ (あるいは、$\{A_k\}_{k=1,2,\cdots,d}$) が Ω 上の coherent configuration であるとは、

CC1. $\sum_{k=1}^{d} A_k = J$ all 1 行列

CC2. ある $r < d$ に対して、$\sum_{k=1}^{r} A_k = I$ 単位行列

CC3. ${}^tA_k = A_{k'}$

CC4. $A_iA_j = \sum_{k=1}^{d} p_{i,j,k}A_k$

その特別な場合として、アソシエーションスキーム $\Longleftrightarrow r = 1$ すなわち $A_1 = I$ が定義される。上の例は、coherent configuration になっている。

2　置換群との関係

G を Ω 上の置換群とし、G を $\Omega \times \Omega$ 上への作用を下で定める。

$$g \in G,\quad (\alpha,\beta) \in \Omega \times \Omega \text{ に対して} \quad (\alpha,\beta)^g = (\alpha^g, \beta^g)$$

このとき、$\{R_1, R_2, \cdots, R_d\}$ を G のこの作用による orbit の全体とすると、$\{R_k\}_{k=1,2,\cdots,d}$ は coherent configuration になる。ただし、$1 \le k \le r$ で $R_k \subseteq \{(\alpha,\alpha)|\alpha \in \Omega\}$ となるように並べる。特に、G が Ω 上可移のとき、$R_1 = \{(\alpha,\alpha)|\alpha \in \Omega\}$ となり、$\{R_k\}_{k=1,2,\cdots,d}$ はアソシエーションスキームとなる。

アソシエーションスキームの同型計算は、サイズが小さい場合のアソシエーションスキームの同型類の分類で必要であったが、置換群の正規化群計算への応用、おもに、可移な群の場合で役に立った。coherent configuration の同型計算は可移でない群での正規化群計算に使える。

例　$\Omega = \{1,2,3,4,5,6\}$、G =Group$((1,2,3,4)(5,6),(1,3))$ とする。G の orbit は $\{1,2,3,4\}$ と $\{5,6\}$ で、

coherent configuration は　$A = \begin{pmatrix} 1 & 3 & 4 & 3 & 5 & 7 \\ 3 & 1 & 3 & 4 & 7 & 5 \\ 4 & 3 & 1 & 3 & 5 & 7 \\ 3 & 4 & 3 & 1 & 7 & 5 \\ 6 & 8 & 6 & 8 & 2 & 9 \\ 8 & 6 & 8 & 6 & 9 & 2 \end{pmatrix}$ で与えられる。

アソシエーションスキームしか使わない方法では、

$\begin{pmatrix} 1 & 3 & 4 & 3 & 5 & 5 \\ 3 & 1 & 3 & 4 & 5 & 5 \\ 4 & 3 & 1 & 3 & 5 & 5 \\ 3 & 4 & 3 & 1 & 5 & 5 \\ 6 & 6 & 6 & 6 & 2 & 7 \\ 6 & 6 & 6 & 6 & 7 & 2 \end{pmatrix}$ として、

orbit ごとにアソシエーションスキームを作って、その直和を考えていた。この行列は、直積群 Group$((1,2,3,4))\times$ Group$((5,6))$ の作る coherent configuration になっている。

ソフトウェアパッケージ CoCo について (http://www.win.tue.nl/ aeb/)

CoCo は、coherent configuration に関する計算パッケージであって、

- G は生成元と基本関係で定義、部分群 H の coset による置換群 G/H の計算
- 置換群の様々な置換表現の計算
- 置換表現から作られる coherent configuration の計算

- $p_{i,j,k}$ の計算
- sub-configuration (subscheme/fusion scheme/merged scheme ともいう) の計算
- それらの自己同型群の位数の計算

を行う。ただし、CoCo で計算する自己同型は R_k, $k = 1, 2, \cdots, d$ は動かさない。

同型な coherent configuration の例　$\Omega = \{1, 2, 3, 4, 5, 6\}$

$$A = \begin{pmatrix} 1 & 3 & 4 & 5 & 5 & 5 \\ 4 & 1 & 3 & 5 & 5 & 5 \\ 3 & 4 & 1 & 5 & 5 & 5 \\ 6 & 6 & 6 & 2 & 7 & 8 \\ 6 & 6 & 6 & 8 & 2 & 7 \\ 6 & 6 & 6 & 7 & 8 & 2 \end{pmatrix} \quad B = \begin{pmatrix} 1 & 7 & 8 & 5 & 5 & 5 \\ 8 & 1 & 7 & 5 & 5 & 5 \\ 7 & 8 & 1 & 5 & 5 & 5 \\ 6 & 6 & 6 & 2 & 3 & 4 \\ 6 & 6 & 6 & 4 & 2 & 3 \\ 6 & 6 & 6 & 3 & 4 & 2 \end{pmatrix} \quad C = \begin{pmatrix} 2 & 7 & 8 & 6 & 6 & 6 \\ 8 & 2 & 7 & 6 & 6 & 6 \\ 7 & 8 & 2 & 6 & 6 & 6 \\ 5 & 5 & 5 & 1 & 3 & 4 \\ 5 & 5 & 5 & 4 & 1 & 3 \\ 5 & 5 & 5 & 3 & 4 & 1 \end{pmatrix}$$

$A \longrightarrow B$ の同型写像：$R_3 \leftrightarrow R_7$,　$R_4 \leftrightarrow R_8$　(行列成分の値の変換)

$A \longrightarrow C$ の同型写像：$1 \leftrightarrow 4$,　$2 \leftrightarrow 5$,　$3 \leftrightarrow 6$ (行および列の置換，つまり，$\Omega = \{1, 2, 3, 4, 5, 6\}$ の点の置換)

coherent configuration の同型計算プログラムの計算手順

(アソシエーションスキームの同型計算プログラムを修正して作った。)

同型となる可能性のある $\{R_1, R_2, \cdots, R_d\}$ の置換の計算

　←　$p_{i,j,k}$ の table に対する計算

$\{R_1, R_2, \cdots, R_d\}$ のどれも固定する同型の計算

　←　行列 A の行および列の置換の計算 (こちらは簡単だった)

3　計算実験

3.1　置換群の正規化群計算

群 G は 20 次から 30 次までの可移な置換群 TransitiveGroup(n, k) の対称群 $Sym(n)$ の中での G の正規化群を計算した。プログラムは GAP システムを使って書いた。下の 3 通りの計算の比較を行った。

- GAP4 — GAP システムで用意されている Normalizer in $Sym(n)$ and $Alt(n)$.
- ISSAC00 — アソシエーションスキームの同型計算を応用したプログラム [1]
- CCNorm06 — coherent configuration の同型計算を応用した今回のプログラム

時間区分内で計算できた群の個数

時間	GAP4	ISSAC00	CCNorm06
$*\leq$0.1 秒	10510	1829	48
0.1 秒 $<*\leq$0.2 秒	11728	7231	558
0.2 秒 $<*\leq$0.5 秒	5433	22898	9518
0.5 秒 $<*\leq$1 秒	2200	2973	12235
1 秒 $<*\leq$2 秒	1098	629	7727
2 秒 $<*\leq$5 秒	1015	363	4539
5 秒 $<*\leq$10 秒	621	182	946
10 秒 $<*\leq$30 秒	834	232	733
30 秒 $<*\leq$1 分	381	126	129
1 分 $<*\leq$2 分	480	40	43
2 分 $<*\leq$5 分	486	30	28
5 分 $<*\leq$10 分	357	6	25
10 分 $<*\leq$30 分	348	9	19
30 分 $<*\leq$1 時間	114	12	10
1 時間 $<*\leq$2 時間	63	15	11
2 時間 $<*\leq$5 時間	112	24	27
5 時間 $<*\leq$10 時間	85	7	9
10 時間 $<*$	755	14	15

正規化群計算 Norm($Sym(n)$,TransitiveGroup(n,k)) におけるいくつかの例における計算時間 (秒)

n	k	CCNorm06	ISSAC00	n	k	CCNorm06	ISSAC00
24	2828	0.8	134	27	1799	10755	0.5
24	7234	0.6	157	27	1822	11160	0.5
27	333	0.5	147	28	413	813	6.5
28	160	1.2	4718	28	488	961	5.8
28	238	0.9	2877	30	4898	13742	37.7
28	273	1.7	6210	30	4899	55084	6.8
28	346	2.6	5270				
30	143	4.1	416				
30	1721	4.7	379				

TransitiveGroup(27, 1799) について
正規化群計算の途中にでてくる群の orbit $\{\{3,7\},\{4,8\},\{5,6\},\{10,16,11,15,17,12,13,18,14\},$ $\{19,24,27,20,23,25,26,22,21\}\}$ この群の長さ 9 の orbit 上の作用は、E(9):2D_8 位数 144, と M(9)=E(9):Q_8 位数 72, で異なっているが、ともに 2 重可移になっている。ISSAC00 ではアソシエーションスキームの直和を作るとき，位数も見ているので、正規化群はこの 2 つの orbit を固定することが、あらかじめ分かる。しかし、CCNorm06 で、coherent configuration を作ったとき下の様になり、左上の 9×9 行列と右下の 9×9 行列を入れ換える同型が存在して、この 2 つの orbit は区別できなくなる。

$$\begin{pmatrix}
1 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
3 & 1 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
3 & 3 & 1 & 3 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
3 & 3 & 3 & 1 & 3 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
3 & 3 & 3 & 3 & 1 & 3 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
3 & 3 & 3 & 3 & 3 & 1 & 3 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
3 & 3 & 3 & 3 & 3 & 3 & 1 & 3 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 1 & 3 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
3 & 3 & 3 & 3 & 3 & 3 & 3 & 3 & 1 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 & 4 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 2 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 6 & 2 & 6 & 6 & 6 & 6 & 6 & 6 & 6 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 6 & 6 & 2 & 6 & 6 & 6 & 6 & 6 & 6 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 6 & 6 & 6 & 2 & 6 & 6 & 6 & 6 & 6 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 6 & 6 & 6 & 6 & 2 & 6 & 6 & 6 & 6 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 6 & 6 & 6 & 6 & 6 & 2 & 6 & 6 & 6 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 6 & 6 & 6 & 6 & 6 & 6 & 2 & 6 & 6 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 2 & 6 \\
5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 5 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 6 & 2
\end{pmatrix}$$

3.2 可移置換群の block system からできる coherent configuration

20 次から 30 次までの可移な置換群から選んだ block system を固定する部分群の作る coherent configuration の (R_k をの置換を許す) 自己同型群の計算

選び方：自己同型群がレス積群にならないもの　5157 個

計算時間：最大 40 秒程度　合計：約 8000 秒，　平均：1.6 秒

時間区分内で計算できた configuration の個数

$*$ ≤0.1 秒	1
0.1 秒 $<*\leq$ 0.2 秒	16
0.2 秒 $<*\leq$ 0.5 秒	1486
0.5 秒 $<*\leq$ 1 秒	1983
1 秒 $<*\leq$ 2 秒	1044
2 秒 $<*\leq$ 5 秒	61
5 秒 $<*\leq$ 10 秒	490
10 秒 $<*\leq$ 30 秒	75
30 秒 $<*\leq$ 1 分	1

4 GAP システムの置換群における正規化群計算

GAP システムでは、関数 AutomorphismGroupPermGroup(下の表では AutoPerm と略記) が置換群の正規化群の計算を直接している．関数 Normalizer は，対称群と交代群の中で計算するときは，適当な部分群 W が存在するときは，それをを求めた後，AutomorphismGroupPermGroup を呼び出して，W の中で正規化群を計算している．この群 W はレス積群となっている。しかし，これが返って計算を遅くしている

ことがある．表において、Isom-Norm は G の作る coherent configuration の自己同型群を計算して、その中で G の正規化群を計算する方法。CCNorm06 では、計算途中でいろいろな coherent configuration を使っているので、置換群の次数が大きくなると memory space 超過が起きてしまう。

Normalizer と AutoPerm と Isom-Norm の比較
G =Stabilizer(PrimitiveGroup$(n,i),n)$ の $Sym(n-1)$ における正規化群計算 (単位：秒)

n	i	Normalizer	AutoPerm	Isom-Norm
81	123	37	0.2	2.0
100	3	77	0.3	2.9
105	9	12	0.3	2.6
112	1	19652	0.4	4.5
120	12	46	0.5	4.1

5 改善すべき点

アソシエーションと比較して coherent configuration では、R_k, $k=1,2,\cdots,d$, の個数 d は非常に大きくなり得る (下の行列参照)。$p_{i,j,k}$ は d^3 個あって、現状では、アソシエーションスキームの同型計算プログラムでそうなっている関係で、d^3 の 3 次元配列 (行列) を使って記録している。

しかし、d が大きいほど、$p_{i,j,k}=0$ が多くの場合で成立するので、GAP や CoCo では、algebra の構造定数を (i,j) 成分が $p_{i,j,k}\neq 0$ となる k のリストと $p_{i,j,k}$ のリストの 2 つのリストからなるリストで記録している。

```
[ [ 1,12,13,13,15,15,17,17,19,19,21,21,23,23,25,25,27,27,29,29,31,31 ],
  [12, 1,13,13,15,15,17,17,19,19,21,21,23,23,25,25,27,27,29,29,31,31 ],
  [14,14, 2,33,34,34,36,36,38,38,40,40,42,42,44,44,46,46,48,48,50,50 ],
  [14,14,33, 2,34,34,36,36,38,38,40,40,42,42,44,44,46,46,48,48,50,50 ],
  [16,16,35,35, 3,52,53,53,55,55,57,57,59,59,61,61,63,63,65,65,67,67 ],
  [16,16,35,35,52, 3,53,53,55,55,57,57,59,59,61,61,63,63,65,65,67,67 ],
  [18,18,37,37,54,54, 4,69,70,70,72,72,74,74,76,78,80,80,82,82,84,84 ],
  [18,18,37,37,54,54,69, 4,70,70,72,72,74,74,78,76,80,80,82,82,84,84 ],
     .... 中略....
  [30,30,49,49,66,66,83,83,96,96,107,107,116,116,123,123,128,128,10,131,132,132 ],
  [30,30,49,49,66,66,83,83,96,96,107,107,116,116,123,123,128,128,131,10,132,132 ],
  [32,32,51,51,68,68,85,85,98,98,109,109,118,118,125,125,130,130,133,133,11,134 ],
  [32,32,51,51,68,68,85,85,98,98,109,109,118,118,125,125,130,130,133,133,134,11 ] ],
```

参 考 文 献

[1] I. Miyamoto: Computing normalizers of permutation groups efficiently using isomorphisms of association schemes. In *Proceedings of the 2000 International Symposium on Symbolic and Algebraic Computation*, pp 200–204, C. Traverso, ed. ACM, 2000.

[2] 宮本泉: ブロックシステムを利用した可移な置換群の正規化群計算の高速化. 数式処理 *J.JSSAC*, pp21–23, 13-1 2006.